

APPENDIX B

Document Type Definition (DTD)

The following is a DTD for IMS transaction definitions according to an embodiment of the invention.

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<!-- XMI Automatic DTD Generation -->
```

```
<!-- Metamodel: macro -->
```

```
<!-- _____ -->
```

```
<!-- _____ -->
```

```
<!-- XMI is the top-level XML element for XMI transfer text -->
```

```
<!-- _____ -->
```

```
<!ELEMENT XMI (XMI.header, XMI.content?, XMI.difference*,  
XMI.extensions*) >
```

```
<!ATTLIST XMI  
xmi.version CDATA #FIXED "1.0"  
timestamp CDATA #IMPLIED  
verified (true | false) #IMPLIED
```

```
>  
  
<!-- _____ -->
```

```
<!-- _____ -->
```

```
<!-- XMI.header contains documentation and identifies the model,  
metamodel, and metamodel -->
```

```
<!-- _____ -->
```

```
<!-- _____ -->
```

```
<!ELEMENT XMI.header (XMI.documentation?, XMI.model*, XMI.metamodel*,  
XMI.metamodel*) >
```

```
<!-- _____ -->
```

```
<!-- _____ -->
```

```
<!-- documentation for transfer data -->
```

```
<!-- _____ -->
```

```
<!ELEMENT XMI.documentation (#PCDATA | XMI.owner | XMI.contact |  
XMI.longDescription | XMI.shortDescription |  
XMI.exporter | XMI.exporterVersion |  
XMI.notice)* >
```

```
<!ELEMENT XMI.owner ANY >
```

```

1  <!ELEMENT XMI.contact ANY >
2
3  <!ELEMENT XMI.longDescription ANY >
4
5  <!ELEMENT XMI.shortDescription ANY >
6
7  <!ELEMENT XMI.exporter ANY >
8
9  <!ELEMENT XMI.exporterVersion ANY >
10
11 <!ELEMENT XMI.exporterID ANY >
12
13 <!ELEMENT XMI.notice ANY >
14
15 <!-- _____ -->
16 <!-- _____ -->
17 <!-- XMI.element.att defines the attributes that each XML element
18 <!-- that corresponds to a metamodel class must have to conform to
19 <!-- the XMI specification.
20 <!-- _____ -->
21
22 <!ENTITY % XMI.element.att
23     'xmi.id ID #IMPLIED xmi.label CDATA #IMPLIED xmi.uuid
24     CDATA #IMPLIED' >
25
26 <!-- _____ -->
27 <!-- _____ -->
28 <!-- XMI.link.att defines the attributes that each XML element that
29 <!-- corresponds to a metamodel class must have to enable it to
30 <!-- function as a simple XLink as well as refer to model
31 <!-- constructs within the same XMI file.
32 <!-- _____ -->
33
34 <!ENTITY % XMI.link.att
35     'xml:link CDATA #IMPLIED inline (true | false) #IMPLIED
36     actuate (show | user) #IMPLIED href CDATA #IMPLIED role
37     CDATA #IMPLIED title CDATA #IMPLIED show (embed | replace
38     | new) #IMPLIED behavior CDATA #IMPLIED xmi.idref IDREF
39     #IMPLIED xmi.uuidref CDATA #IMPLIED' >
40
41 <!-- _____ -->
42 <!-- _____ -->
43 <!-- XMI.model identifies the model(s) being transferred
44 <!-- _____ -->
45
46 <!ELEMENT XMI.model ANY >

```

```

1  <!ATTLIST XMI.model
    %XMI.link.att;
    xmi.name    CDATA #REQUIRED
    xmi.version  CDATA #IMPLIED
2  >
3
4  <!-- _____ -->
5  <!-- XMI.metamodel identifies the metamodel(s) for the transferred
6  <!-- data _____ -->
7  <!-- _____ -->
8
9  <!ELEMENT XMI.metamodel ANY >
10 <!ATTLIST XMI.metamodel
    %XMI.link.att;
    xmi.name    CDATA #REQUIRED
    xmi.version  CDATA #IMPLIED
11 >
12
13 <!-- _____ -->
14 <!-- _____ -->
15 <!-- XMI.metametamodel identifies the metametamodel(s) for the
16 <!-- transferred data _____ -->
17 <!-- _____ -->
18
19 <!ELEMENT XMI.metametamodel ANY >
20 <!ATTLIST XMI.metametamodel
    %XMI.link.att;
    xmi.name    CDATA #REQUIRED
    xmi.version  CDATA #IMPLIED
21 >
22
23 <!-- _____ -->
24 <!-- _____ -->
25 <!-- XMI.content is the actual data being transferred
26 <!-- _____ -->
27
28 <!ELEMENT XMI.content ANY >
29
30 <!-- _____ -->
31 <!-- _____ -->
32 <!-- XMI.extensions contains data to transfer that does not conform
33 <!-- to the metamodel(s) in the header _____ -->
34 <!-- _____ -->
35
36 <!ELEMENT XMI.extensions ANY >

```

```

1  <!ATTLIST XMI.extensions
    xmi.extender CDATA #REQUIRED
2  >
3  <!-- _____ -->
4  <!-- extension contains information related to a specific model -->
5  <!-- construct that is not defined in the metamodel(s) in the header -->
6  <!-- _____ -->
7  <!ELEMENT XMI.extension ANY >
8  <!ATTLIST XMI.extension
    %XMI.element.att;
    %XMI.link.att;
    xmi.extender CDATA #REQUIRED
    xmi.extenderID CDATA #REQUIRED
9  >
10 <!-- _____ -->
11 <!-- XMI.difference holds XML elements representing differences to a -->
12 <!-- base model -->
13 <!-- _____ -->
14 <!ELEMENT XMI.difference (XMI.difference | XMI.delete | XMI.add |
    XMI.replace)* >
15 <!ATTLIST XMI.difference
    %XMI.element.att;
    %XMI.link.att;
16 >
17 <!-- _____ -->
18 <!-- XMI.delete represents a deletion from a base model -->
19 <!-- _____ -->
20 <!ELEMENT XMI.delete EMPTY >
21 <!ATTLIST XMI.delete
    %XMI.element.att;
    %XMI.link.att;
    >
    <!-- _____ -->
    <!-- _____ -->
    <!-- XMI.add represents an addition to a base model -->

```



```

1  <!ELEMENT XMI.unionDiscrim ANY >
2  <!ELEMENT XMI.enum EMPTY >
3  <!ATTLIST XMI.enum
4      xmi.value CDATA #REQUIRED
5  >
6  <!ELEMENT XMI.any ANY >
7  <!ATTLIST XMI.any
8      %XMI.link.att;
9      xmi.type CDATA #IMPLIED
10     xmi.name CDATA #IMPLIED
11 >
12 <!ELEMENT XMI.CorbaTypeCode (XMI.CorbaTcAlias | XMI.CorbaTcStruct |
13     XMI.CorbaTcSequence | XMI.CorbaTcArray |
14     XMI.CorbaTcEnum | XMI.CorbaTcUnion |
15     XMI.CorbaTcExcept | XMI.CorbaTcString |
16     XMI.CorbaTcWstring | XMI.CorbaTcShort |
17     XMI.CorbaTcLong | XMI.CorbaTcUshort |
18     XMI.CorbaTcUlong | XMI.CorbaTcFloat |
19     XMI.CorbaTcDouble | XMI.CorbaTcBoolean |
20     XMI.CorbaTcChar | XMI.CorbaTcWchar |
21     XMI.CorbaTcOctet | XMI.CorbaTcAny |
22     XMI.CorbaTcTypeCode | XMI.CorbaTcPrincipal |
23     XMI.CorbaTcNull | XMI.CorbaTcVoid |
24     XMI.CorbaTcLongLong |
25     XMI.CorbaTcLongDouble) >
26 <!ATTLIST XMI.CorbaTypeCode
27     %XMI.element.att;
28 >
29 <!ELEMENT XMI.CorbaTcAlias (XMI.CorbaTypeCode) >
30 <!ATTLIST XMI.CorbaTcAlias
31     xmi.tcName CDATA #REQUIRED
32     xmi.tcId CDATA #IMPLIED
33 >
34 <!ELEMENT XMI.CorbaTcStruct (XMI.CorbaTcField)* >
35 <!ATTLIST XMI.CorbaTcStruct
36     xmi.tcName CDATA #REQUIRED
37     xmi.tcId CDATA #IMPLIED
38 >
39 <!ELEMENT XMI.CorbaTcField (XMI.CorbaTypeCode) >

```

- Page 38 -

1 <!ELEMENT XMI.CorbaTcExcept (XMI.CorbaTcField)* >
2 <!ATTLIST XMI.CorbaTcExcept
xmi.tcName CDATA #REQUIRED
xmi.tcId CDATA #IMPLIED
3 >
4 <!ELEMENT XMI.CorbaTcString EMPTY >
5 <!ATTLIST XMI.CorbaTcString
xmi.tcLength CDATA #REQUIRED
6 >
7 <!ELEMENT XMI.CorbaTcWstring EMPTY >
8 <!ATTLIST XMI.CorbaTcWstring
xmi.tcLength CDATA #REQUIRED
9 >
10 <!ELEMENT XMI.CorbaTcFixed EMPTY >
11 <!ATTLIST XMI.CorbaTcFixed
xmi.tcDigits CDATA #REQUIRED
xmi.tcScale CDATA #REQUIRED
12 >
13 <!ELEMENT XMI.CorbaTcShort EMPTY >
14 <!ELEMENT XMI.CorbaTcLong EMPTY >
15 <!ELEMENT XMI.CorbaTcUshort EMPTY >
16 <!ELEMENT XMI.CorbaTcUlong EMPTY >
17 <!ELEMENT XMI.CorbaTcFloat EMPTY >
18 <!ELEMENT XMI.CorbaTcDouble EMPTY >
19 <!ELEMENT XMI.CorbaTcBoolean EMPTY >
20 <!ELEMENT XMI.CorbaTcChar EMPTY >
21 <!ELEMENT XMI.CorbaTcWchar EMPTY >
22 <!ELEMENT XMI.CorbaTcOctet EMPTY >
23 <!ELEMENT XMI.CorbaTcAny EMPTY >
24 <!ELEMENT XMI.CorbaTcTypeCode EMPTY >

ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```

1  <![ELEMENT XMI.CorbaTcPrincipal EMPTY >
2  <![ELEMENT XMI.CorbaTcNull EMPTY >
3  <![ELEMENT XMI.CorbaTcVoid EMPTY >
4  <![ELEMENT XMI.CorbaTcLongLong EMPTY >
5  <![ELEMENT XMI.CorbaTcLongDouble EMPTY >
6  <!-- _____ -->
7  <!-- METAMODEL: macro _____ -->
8  <!-- _____ -->
9  <!-- METAMODEL CLASS: ApplicationControlMacro _____ -->
10 <!-- _____ -->
11 <![ELEMENT ApplicationControlMacro.ResidencyOption EMPTY >
12 <![ATTLIST ApplicationControlMacro.ResidencyOption
    xmi.value ( Resident | DynamicOption ) #REQUIRED
13 >
14 <![ELEMENT ApplicationControlMacro.isFPath EMPTY >
15 <![ATTLIST ApplicationControlMacro.isFPath
    xmi.value ( False | True ) #REQUIRED
16 >
17 <![ELEMENT ApplicationControlMacro.FPathBufferSize (#PCDATA |
    XMI.reference)* >
18 <![ELEMENT ApplicationControlMacro.GeneratedPSBName (#PCDATA |
    XMI.reference)* >
19 <![ELEMENT ApplicationControlMacro.GeneratedPSBLanguage EMPTY >
20 <![ATTLIST ApplicationControlMacro.GeneratedPSBLanguage
    xmi.value ( Assembler | COBOL | PL1 | PLI | Pascal ) #REQUIRED
21 >
22 <![ELEMENT ApplicationControlMacro.ProgramType EMPTY >
23 <![ATTLIST ApplicationControlMacro.ProgramType
    xmi.value ( TeleProcessing | Batch ) #REQUIRED

```



```

1  <!ELEMENT TransactionControlMacro.isDCLogWriteAhead EMPTY >
2  <!ATTLIST TransactionControlMacro.isDCLogWriteAhead
   xmi.value ( False | True ) #REQUIRED
3  >
4  <!ELEMENT TransactionControlMacro.toUpperCase EMPTY >
5  <!ATTLIST TransactionControlMacro.toUpperCase
   xmi.value ( False | True ) #REQUIRED
6  >
7  <!ELEMENT TransactionControlMacro.EditRoutine ( #PCDATA |
8  XMI.reference ) * >
9
10 <!ELEMENT TransactionControlMacro.isFPath EMPTY >
11 <!ATTLIST TransactionControlMacro.isFPath
12 xmi.value ( False | True ) #REQUIRED
13 >
14 <!ELEMENT TransactionControlMacro.FPathBufferSize ( #PCDATA |
15 XMI.reference ) * >
16
17 <!ELEMENT TransactionControlMacro.isInquiry EMPTY >
18 <!ATTLIST TransactionControlMacro.isInquiry
19 xmi.value ( False | True ) #REQUIRED
20 >
21 <!ELEMENT TransactionControlMacro.isRecover EMPTY >
22 <!ATTLIST TransactionControlMacro.isRecover
23 xmi.value ( False | True ) #REQUIRED
24 >
25 <!ELEMENT TransactionControlMacro.MaxRegions ( #PCDATA |
26 XMI.reference ) * >
27
28 <!ELEMENT TransactionControlMacro.Mode EMPTY >
29 <!ATTLIST TransactionControlMacro.Mode
30 xmi.value ( MultipleMessage | SingleMessage ) #REQUIRED
31 >
32 <!ELEMENT TransactionControlMacro.MessageType EMPTY >
33 <!ATTLIST TransactionControlMacro.MessageType
34 xmi.value ( MultipleSegment | SingleSegment ) #REQUIRED
35 >
36 <!ELEMENT TransactionControlMacro.MessageIsResponse EMPTY >

```

1 <!ATTLIST TransactionControlMacro.MessageIsResponse
xmi.value (Response | NonResponse) #REQUIRED
2 >
3 <!ELEMENT TransactionControlMacro.MessageClass (#PCDATA |
XMI.reference)* >
4 <!ELEMENT TransactionControlMacro.ParallelLimit (#PCDATA |
XMI.reference)* >
5 <!ELEMENT TransactionControlMacro.ProcessLimitCount (#PCDATA |
6 XMI.reference)* >
7 <!ELEMENT TransactionControlMacro.ProcessLimitSeconds (#PCDATA |
XMI.reference)* >
8 <!ELEMENT TransactionControlMacro.PriorityNormal (#PCDATA |
XMI.reference)* >
9 <!ELEMENT TransactionControlMacro.PriorityLimit (#PCDATA |
10 XMI.reference)* >
11 <!ELEMENT TransactionControlMacro.PriorityLimitCount (#PCDATA |
XMI.reference)* >
12
13 <!ELEMENT TransactionControlMacro.isRouting EMPTY >
<!ATTLIST TransactionControlMacro.isRouting
xmi.value (False | True) #REQUIRED
14 >
15 <!ELEMENT TransactionControlMacro.ScheduleOption (#PCDATA |
XMI.reference)* >
16 <!ELEMENT TransactionControlMacro.SegmentNumber (#PCDATA |
17 XMI.reference)* >
18 <!ELEMENT TransactionControlMacro.SegmentSize (#PCDATA |
XMI.reference)* >
19 <!ELEMENT TransactionControlMacro.isSerial EMPTY >
<!ATTLIST TransactionControlMacro.isSerial
20 xmi.value (False | True) #REQUIRED
>
21 <!ELEMENT TransactionControlMacro.SPASize (#PCDATA | XMI.reference)* >

1 TransactionControlMacro.SysIDLocal?,
2 TransactionControlMacro.isWaitForInput?,
3 XMI.extension*,
4 TransactionControlMacro.APPLCTN?)? >
5 <!ATTLIST TransactionControlMacro
6 %XMI.element.att;
7 %XMI.link.att;
8 >
9
10 <!ELEMENT macro ((ApplicationControlMacro | TransactionControlMacro)*) >
11 <!ATTLIST macro
12 %XMI.element.att;
13 %XMI.link.att;
14 >
15
16
17
18
19
20
21